In cases where there is disaster recovery, if data was transmitted to persistent storage **170**, a backup physical host can read all the data from the persistent storage and rebuild the memory space for the virtual machine.

One advantage of using a hardware MSM is that it can still run after a power outage. The MSM can also quickly obtain hash values and transmit the hash values increasing the likelihood a backup virtual machine can start successfully. Using hash values also reduces the possibility of data corruption. The MSM can also keep track of different virtual machines so that synchronizing can be done to different backup servers for load balancing. The backup memory can further allow data recovery or even forensic research.

In some embodiments, pages that are resident in memory are synchronized, while pages in the page cache or other caches are not synchronized. Additionally, memory mapped regions for hardware and files need not by synchronized.

Although the operations of some of the disclosed methods are described in a particular, sequential order for convenient presentation, it should be understood that this manner of description encompasses rearrangement, unless a particular ordering is required by specific language set forth below. For example, operations described sequentially may in some cases be rearranged or performed concurrently. Moreover, for the sake of simplicity, the attached figures may not show the various ways in which the disclosed methods can be used in conjunction with other methods.

Any of the disclosed methods can be implemented as computer-executable instructions stored on one or more computer-readable storage media (e.g., non-transitory computer-readable media, such as one or more optical media discs, volatile memory components (such as DRAM or SRAM), or nonvolatile memory components (such as flash memory or hard drives)) and executed on a computer. As should be readily understood, the term computer-readable storage media does not include communication connections, such as modulated data signals.

For clarity, only certain selected aspects of the software-based implementations are described. Other details that are well known in the art are omitted. For example, it should be understood that the disclosed technology is not limited to any specific computer language or program. For instance, the disclosed technology can be implemented by software written in C++, Java, Perl, JavaScript, Adobe Flash, or any other suitable programming language. Likewise, the disclosed technology is not limited to any particular computer or type of hardware. Certain details of suitable computers and hardware are well known and need not be set forth in detail in this disclosure.

It should also be well understood that any functionality described herein can be performed, at least in part, by one or more hardware logic components, instead of software. For example, and without limitation, illustrative types of hardware logic components that can be used include Field-programmable Gate Arrays (FPGAs), Program-specific Integrated Circuits (ASICs), Program-specific Standard Products (ASSPs), System-on-a-chip systems (SOCs), Complex Programmable Logic Devices (CPLDs), etc.

Furthermore, any of the software-based embodiments (comprising, for example, computer-executable instructions for causing a computer to perform any of the disclosed methods) can be uploaded, downloaded, or remotely accessed through a suitable communication means. Such suitable communication means include, for example, the Internet, the World Wide Web, an Intranet, software applications, cable (including fiber optic cable), magnetic communications, electromagnetic communications (including RF, microwave, and infrared communications), electronic communications, or other such communication means.

The disclosed methods, apparatus, and systems should not be construed as limiting in any way. Instead, the present disclosure is directed toward all novel and nonobvious features and aspects of the various disclosed embodiments, alone and in various combinations and subcombinations with one another. The disclosed methods, apparatus, and systems are not limited to any specific aspect or feature or combination thereof, nor do the disclosed embodiments require that any one or more specific advantages be present or problems be solved.

In view of the many possible embodiments to which the principles of the disclosed invention may be applied, it should be recognized that the illustrated embodiments are only preferred examples of the invention and should not be taken as limiting the scope of the invention. Rather, the scope of the invention is defined by the following claims. We therefore claim as our invention all that comes within the scope of these claims.

What is claimed is:

1. A method of tracking memory changes for high availability virtual machines, comprising:

providing a primary virtual machine instantiated on a first physical host;

providing a failover virtual machine instantiated on a second physical host, the failover virtual machine for being switched to upon failure or abnormal termination of the primary virtual machine;

receiving data associated with a memory write request from the primary virtual machine in a first hardware memory synchronization manager;

in response to the memory write request, writing to a first memory associated with the primary virtual machine and transmitting, over a network, at least the data to a second hardware memory synchronization manager located on the second physical host, wherein the first hardware memory synchronization manager initiates both the writing to the first memory and the transmitting the data over the network; and

writing the data to a second memory associated with the failover virtual machine so as to maintain substantially redundant copies between the first and second memories.

2. The method of claim **1**, wherein the first and second hardware memory synchronization managers maintain a memory map for their respective primary and failover virtual machines.

3. The method of claim **1**, wherein the primary virtual machine has an identification associated therewith that is maintained in the first hardware memory synchronization manager and that is included in the memory write request.

4. The method of claim **1**, further including generating a hash value associated with the memory write request and transmitting the hash value to the second hardware memory synchronization manager together with the data.

5. The method of claim **1**, further including, in response to the memory write request, writing the data to persistent storage separate from the first and second physical hosts.

6. A method of tracking memory in virtual machines, comprising:

requesting data to be written to a first memory in a first virtual machine running on a first physical host;

in response to the request, using a hardware memory synchronization manager, writing the data to the first